DSC 462, Homework 4

Kefu Zhu

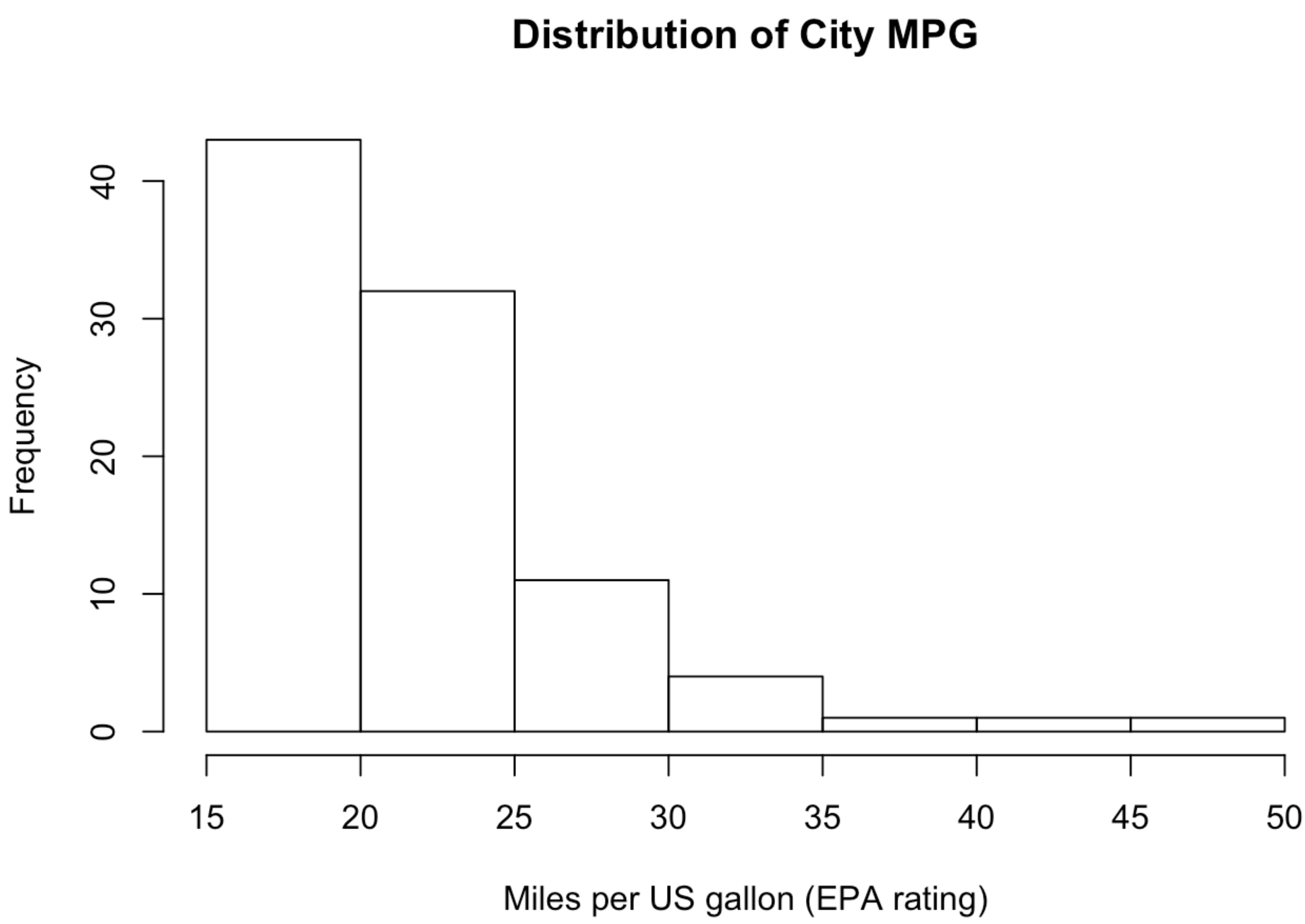
11/21/2018

```r
library(MASS)
```

Question 1

(a)

```r
hist(Cars93$MPG.city,
     main = 'Distribution of City MPG',
     xlab = 'Miles per US gallon (EPA rating)')
```



Answer: It is clear to see the distribution of `MPG.city` is right skewed

(b)

```
empirical_rule = function(x) {
  # Initialize the output 3x2 matrix with zeros
  result = matrix(0,
                  nrow = 3,
                  ncol = 2,
                  dimnames = list(c('within 1 SD',
                                    'within 2 SD',
                                    'within 3 SD'),
                                  c('Sample Proportion (%)',
                                    'Theoretical Proportion (%)'))
                  )
  # Insert the theoretical values
  result[, 'Theoretical Proportion (%)'] = c(68, 95, 99.7)

  # Compute the mean and sample standard deviation
  sample_sd = sd(x)
  sample_mean = mean(x)

  # Compute the sample proportion and insert them into the matrix
  result[1, 'Sample Proportion (%)'] = 100 * sum(abs(x - mean(x)) < 1 * sample_sd) /
length(x)
  result[2, 'Sample Proportion (%)'] = 100 * sum(abs(x - mean(x)) < 2 * sample_sd) /
length(x)
  result[3, 'Sample Proportion (%)'] = 100 * sum(abs(x - mean(x)) < 3 * sample_sd) /
length(x)

  # Return the matrix
  return(result)
}
```

Test on Random Sample

Normal Distribution

```
normal_sample = rnorm(1000)
empirical_rule(normal_sample)
```

```
##             Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD                  70.1                       68.0
## within 2 SD                  95.0                       95.0
## within 3 SD                  99.5                       99.7
```

Exponential Distribution

```
exp_sample = rexp(1000)
empirical_rule(exp_sample)
```

```
##             Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD                  89.0                       68.0
## within 2 SD                  95.8                       95.0
## within 3 SD                  98.4                       99.7
```

Uniform Distribution

```
uniform_sample = runif(1000)
empirical_rule(uniform_sample)
```

```
##             Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD                  57.9                       68.0
## within 2 SD                 100.0                       95.0
## within 3 SD                 100.0                       99.7
```

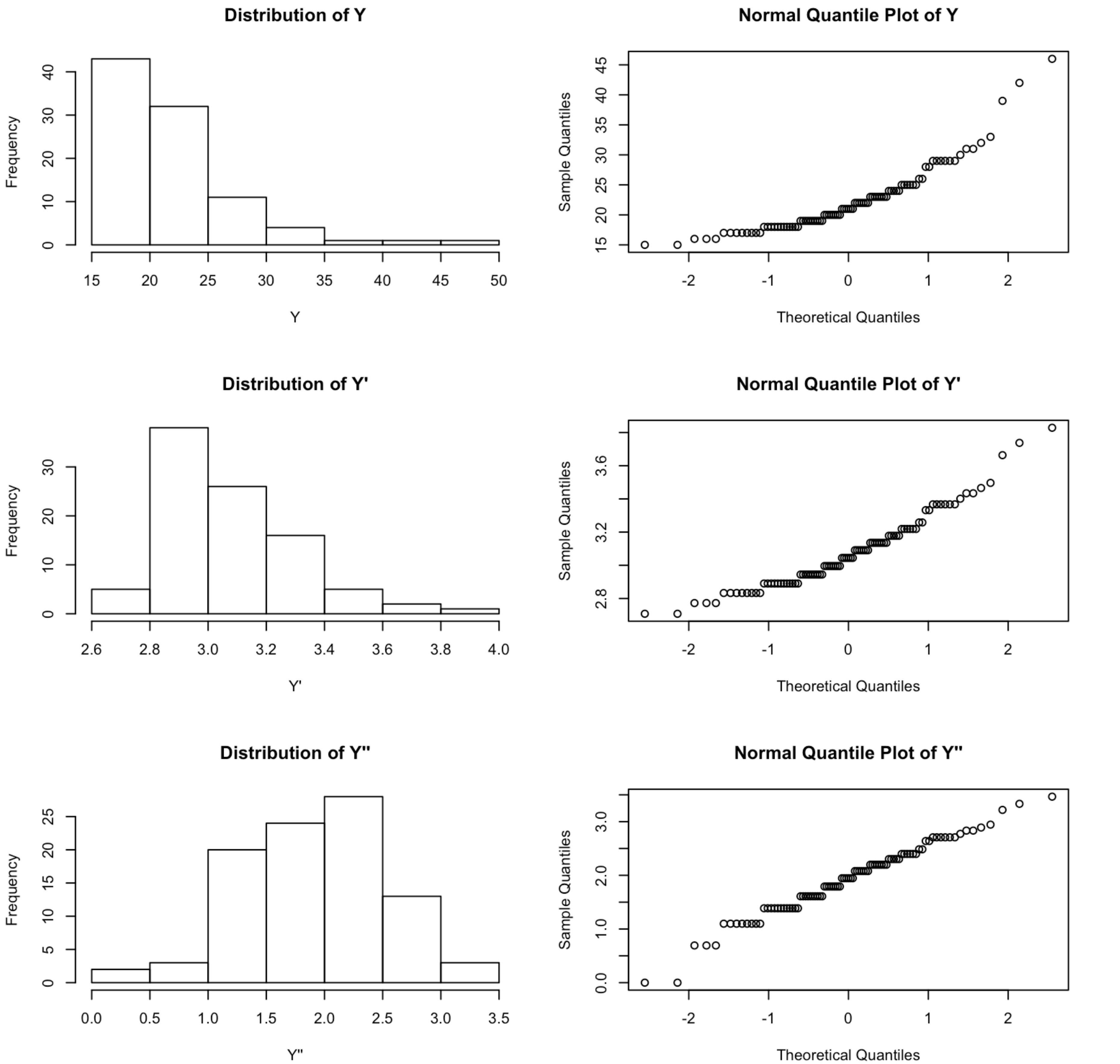
(c)

```
# Set 3x2 plot grid
par(mfrow=c(3,2))
# Set Y
y = Cars93$MPG.city

# Histogram for Y
hist(y, main = 'Distribution of Y', xlab = 'Y')
# Normal Quantile Plot for Y
qqnorm(y, main = 'Normal Quantile Plot of Y')

# Computer Y'
y_prime = log(y)
# Histogram for Y'
hist(y_prime, main = "Distribution of Y'", xlab = "Y'")
# Normal Quantile Plot for Y
qqnorm(y_prime, main = "Normal Quantile Plot of Y'")

# Computer Y''
y_double_prime = log(y - min(y) + 1)
# Histogram for Y'
hist(y_double_prime, main = "Distribution of Y''", xlab = "Y''")
# Normal Quantile Plot for Y
qqnorm(y_double_prime, main = "Normal Quantile Plot of Y''")
```




```
# Reset the plot grid setting
dev.off()
```

```
empirical_rule(y)
```

```
##              Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD               77.41935                       68.0
## within 2 SD               96.77419                       95.0
## within 3 SD               97.84946                       99.7
```

```
empirical_rule(y_prime)
```

```
##              Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD            68.81720                       68.0
## within 2 SD            96.77419                       95.0
## within 3 SD            98.92473                       99.7
```

```
empirical_rule(y_double_prime)
```

```
##              Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD            68.81720                       68.0
## within 2 SD            95.69892                       95.0
## within 3 SD           100.00000                       99.7
```

Answer: Only the transformation, $Y'' = log(Y - min(Y) + 1)$, is close enough to be approximated to normal distribution

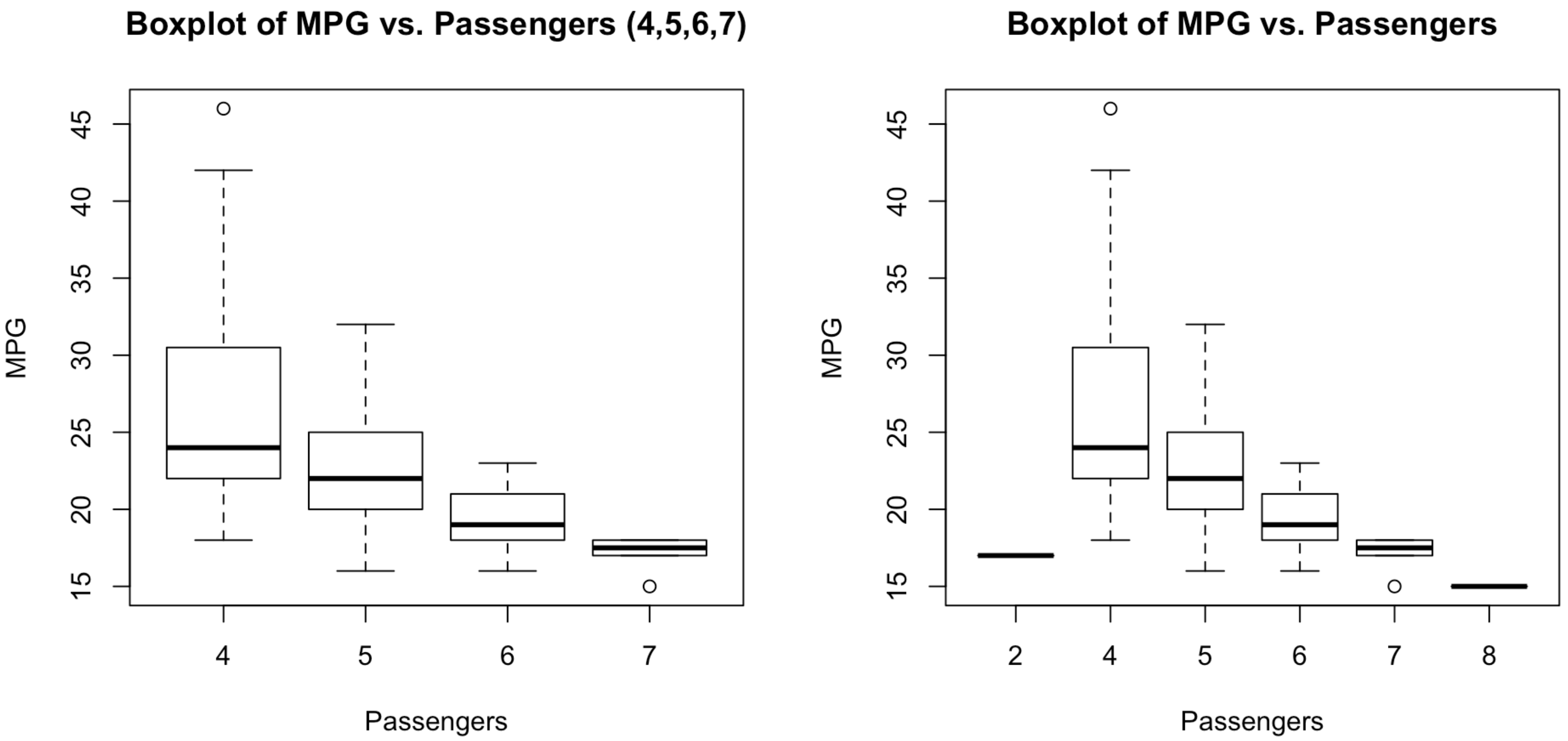
Question 2

(a)

```
# Set 1x2 plot grid
par(mfrow=c(1,2))
# Subset data
newdata = subset(Cars93, Cars93$Passengers %in% c(4,5,6,7))

# Boxplot for Passengers equal to 4,5,6,7
boxplot(
  MPG.city ~ Passengers,
  data = newdata,
  main = 'Boxplot of MPG vs. Passengers (4,5,6,7)',
  xlab = 'Passengers',
  ylab = 'MPG'
  )

# Boxplot for all data
boxplot(
  MPG.city ~ Passengers,
  Cars93,
  main = 'Boxplot of MPG vs. Passengers',
  xlab = 'Passengers',
  ylab = 'MPG'
  )
```




```
# Reset the plot grid setting
dev.off()
```

Answer: From the above two boxplots, most of the data are from cars with passenger capacity of 4,5,6,7. Only few observations are from cars with passenger capacity of 2 or 8 people. In addition, the mean value of MPG value and its variation also varies quite a lot in different groups (Cars that has different passenger capacity).

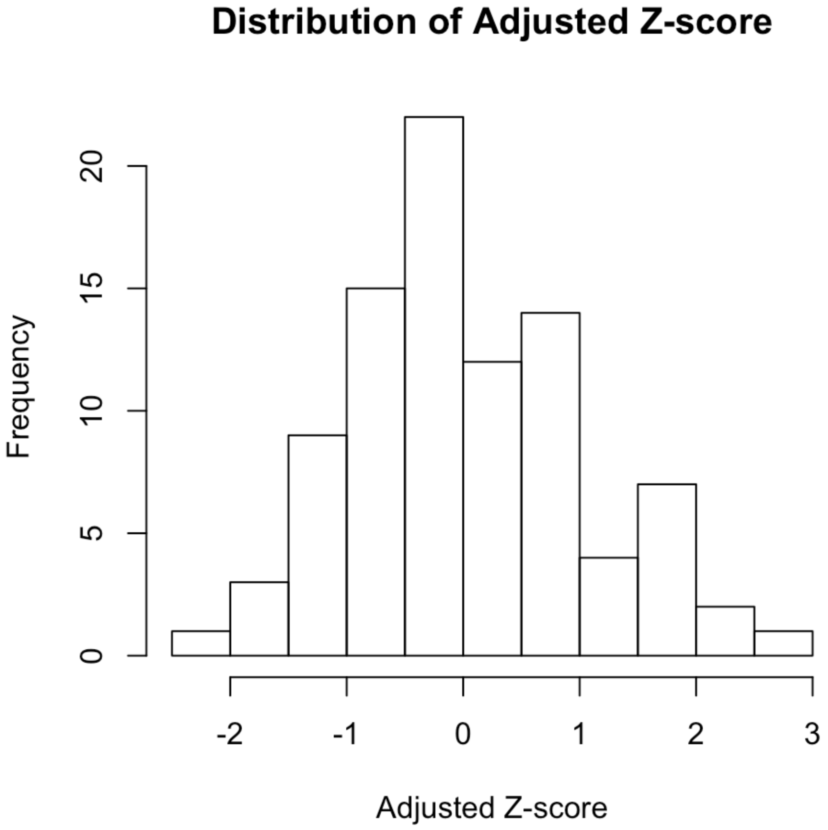
(b)

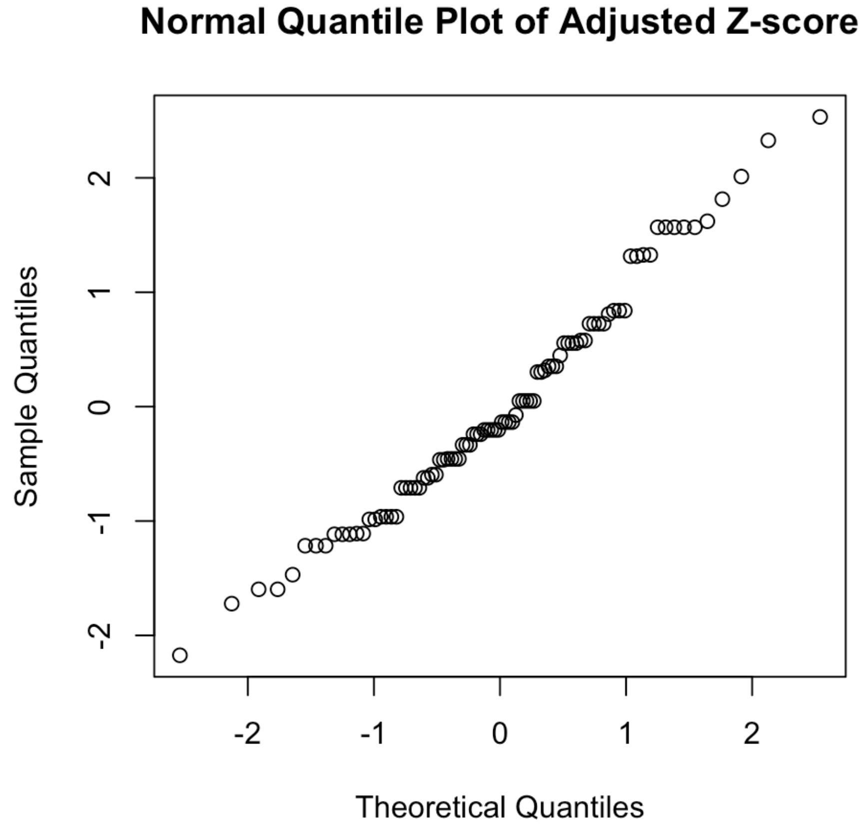
```r
# Compute mean value of MPG for different group
mean_4 = mean(Cars93[Cars93$Passengers == 4,'MPG.city'])
mean_5 = mean(Cars93[Cars93$Passengers == 5,'MPG.city'])
mean_6 = mean(Cars93[Cars93$Passengers == 6,'MPG.city'])
mean_7 = mean(Cars93[Cars93$Passengers == 7,'MPG.city'])
# Compute standard deviation of MPG for different group
sd_4 = sd(Cars93[Cars93$Passengers == 4,'MPG.city'])
sd_5 = sd(Cars93[Cars93$Passengers == 5,'MPG.city'])
sd_6 = sd(Cars93[Cars93$Passengers == 6,'MPG.city'])
sd_7 = sd(Cars93[Cars93$Passengers == 7,'MPG.city'])

# Create an empty vector to store adjusted Z-score
adjust_Z = c()

# Loop through each observation in the dataset and compute adjusted Z-score
for(i in 1:nrow(newdata)){
  # Get the passenger capacity for the current car
  passenger_cap = newdata$Passengers[i]
  # Get the group mean and group standard deviation from its group
  group_mean = get(paste('mean',passenger_cap,sep = '_'))
  group_sd = get(paste('sd',passenger_cap,sep = '_'))
  # Compute the adjusted Z-score for the current car
  z = (newdata$MPG.city[i] - group_mean)/group_sd
  # Insert the adjusted Z-score to the vector
  adjust_Z[i] = z
}
```

```r
# Set 1x2 plot grid
par(mfrow=c(1,2))
# Histogram for adjusted Z-score
hist(adjust_Z, main = "Distribution of Adjusted Z-score", xlab = "Adjusted Z-score")
# Normal Quantile Plot adjusted Z-score
qqnorm(adjust_Z, main = "Normal Quantile Plot of Adjusted Z-score")
```






```
# Reset the plot grid setting
dev.off()
```

```
empirical_rule(adjust_Z)
```

```
##              Sample Proportion (%) Theoretical Proportion (%)
## within 1 SD               67.77778                       68.0
## within 2 SD               95.55556                       95.0
## within 3 SD              100.00000                       99.7
```

Answer: Based on both plots and the result from `empirical_rule()` function, we can say the values Z is approximately normal

Question 3

(a)

```
confidence_interval = function(data, alpha){
  n = length(data)
  mean_ = mean(data)
  sd_ = sd(data)
  z = qt(1-alpha/2, n-1)

  return(c(mean_ - z*sd_/sqrt(n), mean_ + z*sd_/sqrt(n)))
}
```

(b)

Confidence Interval for Passengers = 4

```
# Confidence Interval for Passengers = 4
confidence_interval(data = Cars93[Cars93$Passengers == 4,'MPG.city'], alpha = 0.05)
```

```
## [1] 23.24665 29.88379
```

Confidence Interval for Passengers = 5

```
# Confidence Interval for Passengers = 5
confidence_interval(data = Cars93[Cars93$Passengers == 5,'MPG.city'], alpha = 0.05)
```

```
## [1] 21.55777 24.05199
```

Confidence Interval for Passengers = 6

```
# Confidence Interval for Passengers = 6
confidence_interval(data = Cars93[Cars93$Passengers == 6,'MPG.city'], alpha = 0.05)
```

```
## [1] 18.25713 20.29842
```

Confidence Interval for Passengers = 7

```
# Confidence Interval for Passengers = 7
confidence_interval(data = Cars93[Cars93$Passengers == 7,'MPG.city'], alpha = 0.05)
```

```
## [1] 16.38464 18.11536
```

Answer: Recall that $\bar{X}_4 = 26.5652174$ and $\bar{X}_5 = 22.804878$. Because neither of them is contained in the other's confidence interval. Therefore, confidence intervals for Passengers = 4 and 5 **DO NOT** overlap

(c)

Answer: Yes. To reject the hypothesis $H_o : \mu_1 = \mu_2$ is equivalent to reject $H_m : \mu_1 - \mu_2 = 0$. Since we already know the lower bound of the first confidence interval is larger than the upper bound of the second, then the lower bound of $\mu_1 - \mu_2$

$$min(\mu_1 - \mu_2) = min(\mu_1) - max(\mu_2) > 0.$$

We also know that if a level $1 - \alpha$ confidence interval for a mean doesn't contain 0, we can reject the null hypothesis $H_o : \mu = 0$. Then we can certainly say we now reject $H_m : \mu_1 - \mu_2 = 0 \Leftrightarrow H_o : \mu_1 = \mu_2$

(d)

```
# Two sample t-test
t.test(Cars93[Cars93$Passengers == 4,'MPG.city'],
       Cars93[Cars93$Passengers == 5,'MPG.city'],
       var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  Cars93[Cars93$Passengers == 4, "MPG.city"] and Cars93[Cars93$Passengers ==
5, "MPG.city"]
## t = 2.1926, df = 28.68, p-value = 0.0366
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2510077 7.2696710
## sample estimates:
## mean of x mean of y
##  26.56522  22.80488
```

Answer: Because the p-value is 0.0366, which is smaller than 0.05, we can reject the null hypothesis. The result does not contradict with part(b)

Question 4

(a)

$$H_o : 4 \cdot \mu_4 = 5 \cdot \mu_5$$

$$H_a : 4 \cdot \mu_4 \neq 5 \cdot \mu_5$$

$$T = \frac{n_4 \cdot \bar{X_4} - n_5 \cdot \bar{X_5}}{\sqrt{\frac{16 \cdot S_4^2}{n_4} + \frac{25 \cdot S_5^2}{n_5}}}$$

(b)

```
# Obtain values of PMPG for m = 4, m = 5
PMPG_4 = 4 * Cars93[Cars93$Passengers == 4,'MPG.city']
PMPG_5 = 5 * Cars93[Cars93$Passengers == 5,'MPG.city']

t.test(PMPG_4, PMPG_5, var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  PMPG_4 and PMPG_5
## t = -1.0926, df = 32.447, p-value = 0.2826
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -22.229129   6.702088
## sample estimates:
## mean of x mean of y
##  106.2609  114.0244
```

Answer: Because the p value is 0.2836, which is larger than 0.05, we **FAIL to reject** the null hypothesis. The result is different from part(b) in Question 3

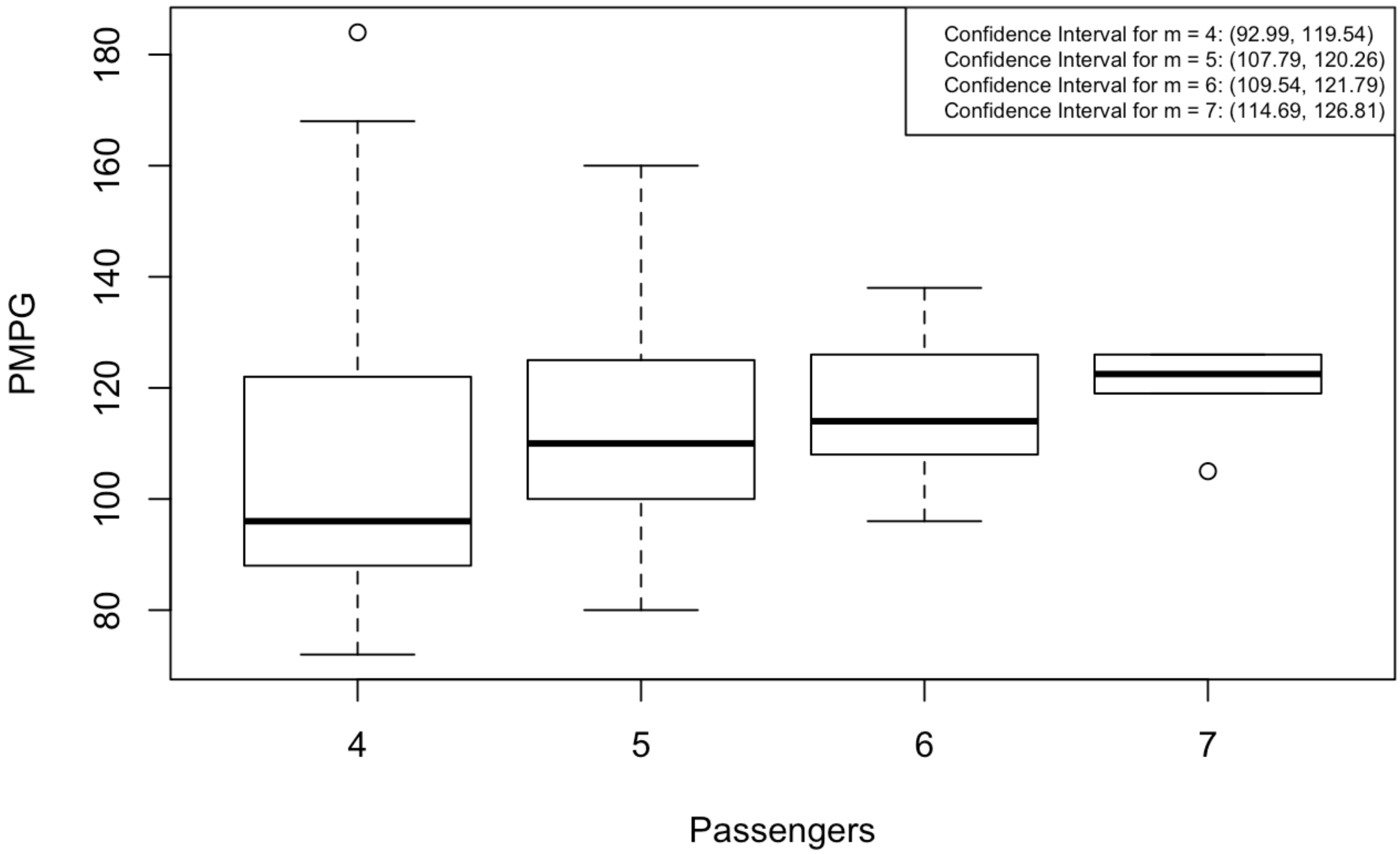
(c)

```
conf_4 = confidence_interval(4*Cars93[Cars93$Passengers == 4,'MPG.city'],0.05)
conf_5 = confidence_interval(5*Cars93[Cars93$Passengers == 5,'MPG.city'],0.05)
conf_6 = confidence_interval(6*Cars93[Cars93$Passengers == 6,'MPG.city'],0.05)
conf_7 = confidence_interval(7*Cars93[Cars93$Passengers == 7,'MPG.city'],0.05)

boxplot(
  Passengers*MPG.city ~ Passengers,
  data = newdata,
  main = 'Boxplot of PMPG vs. Passengers (4,5,6,7)',
  xlab = 'Passengers',
  ylab = 'PMPG'
  )

legend(
  'topright',
  legend = c(
  paste('Confidence Interval for m = 4: (',
        round(conf_4[1],2), ', ',
        round(conf_4[2],2), ')', sep = ''),
  paste('Confidence Interval for m = 5: (',
        round(conf_5[1],2), ', ',
        round(conf_5[2],2), ')', sep = ''),
  paste('Confidence Interval for m = 6: (',
        round(conf_6[1],2), ', ',
        round(conf_6[2],2), ')', sep = ''),
  paste('Confidence Interval for m = 7: (',
        round(conf_7[1],2), ', ',
        round(conf_7[2],2), ')', sep = '')
  ),
  cex = 0.6
)
```

Boxplot of PMPG vs. Passengers (4,5,6,7)



Answer: The means of PMPG seem to be the same between different car classes but the variations within different classes are not the same.